

Linguagem de Programação II

Profº Jânio Eduardo

janio.Eduardo@fbr.edu.br



Roteiro de hoje

- ▶ Universo PHP
- ▶ Sintaxe
- ▶ Operadores
- ▶ Exercícios

- ▶ PHP (um acrônimo recursivo para *PHP: Hypertext Preprocessor*) é uma linguagem de script open source de uso geral, muito utilizada e especialmente guarnecida para o desenvolvimento de aplicações Web embutível dentro do HTML.



Características

- ▶ Velocidade e robustez.
- ▶ Estruturado e orientação a objetos.
- ▶ Portabilidade (independência de plataforma).
- ▶ Tipagem dinâmica.
- ▶ Sintaxe similar a C/C++.
- ▶ Open-source.
- ▶ Server-side.

Histórico – Criador

- ▶ A versão que conhecemos hoje sucede de um produto mais antigo, chamado PHP/FI. Ele foi criado por Rasmus Lerdorf em 1995, inicialmente eram simples scripts Perl, usado para acompanhamento de visitas para seu currículo online.



Histórico

- ▶ Ele nomeou esta série de script de 'Personal Home Page Tools' ou 'PHP Tools'. Como mais funcionalidades foram requeridas, Rasmus escreveu uma implementação C muito maior, que era capaz de comunicar-se com base de dados, e possibilitava à usuários desenvolver simples aplicativos dinâmicos para Web. Ele resolveu disponibilizar o código fonte do PHP/FI para que todos pudessem ver, e também usá-lo, bem como corrigir bugs e melhorar o código e foi aí que toda a brincadeira começou.

Histórico

- ▶ A linguagem é considerada software livre, licenciado sob a PHP License, uma licença incompatível com a GNU General Public License (GPL) devido a restrições no uso do termo PHP.

Linha do tempo PHP

PHP versions timeline



- ▶ Atualmente estimasse que o PHP é usado por 82,6% de todos os sites que utilizam linguagem server-side. Muito dessa popularidade se dar pela grande facilidade no aprendizado e na quantidade de plataformas e ferramentas criadas com a linguagem, como WordPress e Magento (que inclusive utilizamos no desenvolvimento de muitos projetos dos nossos clientes).

Principais clientes



Principais apoiadores



Sintaxe

<https://www.php.net/>

Variáveis

As variáveis no PHP são representadas por um cifrão (\$) seguido pelo nome da variável, os nomes de variável são case-sensitive

```
<?php
```

```
$nome = "Joana Gabrielle";
```

```
$idade = 5;
```

```
echo $nome; # Joana Gabrielle
```

```
echo "\n"; # '\n' é uma quebra de linha
```

```
echo $idade + 1; # 6
```

Examinando o valor de uma variável

- ▶ Para examinarmos o conteúdo de uma variável podemos utilizar a função `print_r()`.
 - `<?php print_r($minha_variavel);`
- ▶ Mas você irá notar que ainda não está legível como você gostaria, então tente incluir umas tags HTML:
 - `<?php`
 - `echo "<pre>";`
 - `print_r($minha_variavel);`
 - `echo "</pre>";`

- ▶ Aí sim você verá o resultado bem formatado. Mas tudo isso de linha só para exibir o valor de uma variável ? Você pode utilizar uma função, por exemplo:

- ```
<?php function my_print_r($minha_variavel) {
 echo "<pre>";
 print_r($minha_variavel);
 echo "</pre>";
}
```

- ▶ Ou utilizar o `var_dump()`. Háaaaa, então porque você não me apresentou logo esse cara ? É porque o `print_r()` é uma função interna, ela acompanha o PHP. O `var_dump()` também é uma função interna, mas para ele funcionar (adequadamente) você precisará ter a extensão x-debug devidamente instalado.
- ▶ `<?php var_dump($minha_variavel);`
  - `var_dump`: exibe tipo e valor, precisa ter o `x-debug instalado
  - `print_r`: exibe valor, fica melhor com a tag `<pre>`



# Funções

- ▶ Funções são uma seção nomeada de código, um pequeno (normalmente pequeno) trecho de código que pode ser chamado de diversos pontos do código.
- ▶ Quando o programador precisar executar um mesmo código várias vezes, ele poderá utilizar-se de funções.
- ▶ As funções introduzem o conceito de reusabilidade, ou seja, “escrever código para depois”.

# Funções

```
▶ <?php
▶ function somar_dois_numeros($param1, $param2)
▶ {
▶ return $param1 + $param2;
▶ }

▶ echo somar_dois_numeros(5, 3);
```

# Operadores aritméticos

- ▶ `<?php`
- ▶ `$a + $b;` # *Adição : Soma de \$a e \$b.*
- ▶ `$a - $b;` # *Subtração : Diferença entre \$a e \$b.*
- ▶ `$a * $b;` # *Multiplicação : Produto de \$a e \$b.*
- ▶ `$a / $b;` # *Divisão : Quociente de \$a e \$b.*
- ▶ `$a % $b;` # *Módulo : Resto de \$a dividido por \$b.*
- ▶ `$a ** $b;` # *Exponencial : Resultado de \$a elevado a \$b.*

# Operadores de Incremento/Decremento

- ▶ `<?php`
- ▶ `++$a;` # Pré-incremento - Incrementa `$a` em um, e então retorna `$a`.
- ▶ `$a++;` # Pós-incremento - Retorna `$a`, e então incrementa `$a` em um.
- ▶ `--$a;` # Pré-decremento - Decrementa `$a` em um, e então retorna `$a`.
- ▶ `$a--;` # Pós-decremento - Retorna `$a`, e então decrementa `$a` em um.

# Operadores Lógicos

- ▶ `<?php`
- ▶ `($a && $b);` # verdadeiro se *\$a* e *\$b* forem verdadeiros.
- ▶ `($a || $b);` # verdadeiro se *\$a* ou *\$b* forem verdadeiros.
- ▶ `($a xor $b);` # verdadeiro se *\$a* ou *\$b* forem verdadeiros, mas se ambos forem verdadeiro, então é falso.

# Como comentar o código

- ▶ *// comentários Linha a Linha*
- ▶ *# comentários Linha a Linha*
- ▶ *// comentários Linha a Linha*
- ▶ **<?php**
  - /\* comentários várias Linhas*
  - comentários várias Linhas*
  - comentários várias Linhas*
  - \*/*

# Estruturas de controle

- ▶ `<?php`
- ▶ `if (expressao-condicional) {`
  - `... executar codigo`
  - `... executar codigo`
  - `... executar codigo }`
- ▶ `else {`
  - `... executar codigo`
  - `... executar codigo`
  - `... executar codigo`
- ▶ `}`

# Laços de repetição - foreach

```
▶ <?php
▶ foreach(['primeiro', 'segundo', 'terceiro'] as $value)
▶ {
▶ echo $value;
▶ } // Exibe... 'primeiro', 'segundo' e 'terceiro'
```



# Laços de repetição – Laço for()

- ▶ `<?php`
- ▶ `for($i = 0; $i < 5, $i++)`
- ▶ `{`
- ▶ `echo $i;`
- ▶ `} // exibe 0, 1, 2, 3, e 4`

# Laços de repetição – Laço while()

- ▶ `<?php`
- ▶ `$i = 0;`
- ▶ `while ($i > 5)`
- ▶ `{`
- ▶ `echo $i; $i++;`
- ▶ `} // exibe 0, 1, 2, 3, e 4`

# Arrays

- ▶ Array é uma estrutura que relaciona valores e chaves, é uma lista de valores armazenados na memória.
- ▶ *<?php # Como se criava array antigamente*  
`$arr = array("primeiro", "segundo", "terceiro");` *# Como se cria um array atualmente*  
`$arr = ["primeiro", "segundo", "terceiro"];`

- ▶ <?php
- ▶ \$arr =
- ▶ [ "cor" => "vermelha",
- ▶ "sabor" => "doce",
- ▶ "formato" => "redonda",
- ▶ "nome" => "maçã" ];

# Objetos

- ▶ `<?php # arquivo index.php`
- ▶ `require "Triangulo.php";`
- ▶ `$objeto = new Triangulo(); #`  
*instanciamos*
- ▶ `$objeto->a = 3; # atribuímos valor`  
`$objeto->b = 4; # idem`
- ▶ `$objeto->c = 5; # idem var_dump(`  
`$objeto->validarForma() );`

```
▶ <?php # arquivo Triangulo.php

▶ class Triangulo
▶ { # suas propriedades
▶ public $a;
▶ public $b;
▶ public $c; # seus métodos
▶ public function validarForma() {
▶ if ($this->a < ($this->b + $this->c)) {
▶ if ($this->b < ($this->a + $this->c)) {
▶ if ($this->c < ($this->a + $this->b)) {
return true; } } } return false; } }
```

Obrigado!!!

Profº Jânio Eduardo

[janio.vasconcellos@gmail.com](mailto:janio.vasconcellos@gmail.com)

(61) 98451-9188